WEST

Help | Logout

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers

Generate Collection

Search Results - Record(s) 1 through 3 of 3 returned.

1. Document ID: US 6070253 A

Entry 1 of 3 File: USPT May 30, 2000

US-PAT-NO: 6070253
DOCUMENT-IDENTIFIER: US 6070253 A
TITLE: Computer diagnostic board that provides system monitoring and permits remote terminal access

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Image

2. Document ID: US 6000040 A

Entry 2 of 3 File: USPT Dec 7, 1999

US-PAT-NO: 6000040
DOCUMENT-IDENTIFIER: US 6000040 A
TITLE: Method and apparatus for diagnosing fault states in a computer system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Image

3. Document ID: US 5864653 A

Entry 3 of 3 File: USPT Jan 26, 1999

US-PAT-NO: 5864653
DOCUMENT-IDENTIFIER: US 5864653 A
TITLE: PCI hot spare capability for failed components

Full | Title | Citation | Front | Review | Classification | Date | Reference | Claims | KWIC | Image

Generate Collection

Terms	Documents
l16 and intercept$3	3

Display 10 Documents including document number 3

Display Format: TI Change Format

Main Menu	Search Form	Posting Counts	Show S Numbers	Edit S Numbers

Help | Logout

WEST

Help | Logout

Main Menu | Search Form | Posting Counts | Show S Numbers | Edit S Numbers

Search Results -

Terms	Documents
l16 and intercept$3	3

Database: US Patents Full-Text Database

Refine Search: l16 and intercept$3

Search History

DB Name	Query	Hit Count	Set Name
USPT	l16 and intercept$3	3	L18
USPT	l16 and intercept$3 near3 call	0	L17
USPT	l15 and test$3	48	L16
USPT	l14 and pci near3 bus	70	L15
USPT	l13 and pci near3 bridge	70	L14
USPT	l1 and cpu	3699	L13
USPT	l8 and interface	1	L12
USPT	l8 and list	0	L11
USPT	l8 and table	0	L10
USPT	l8 and routine	1	L9
USPT	5001712.pn.	1	L8
USPT	l6 and fault	7	L7
USPT	l2 and intercept$3	13	L6
USPT	l2 and intercept$3 near3 mechanism	0	L5
USPT	l2 and intercept$3 near3 access	0	L4
USPT	l2 and intercept$3 near1 access	0	L3
USPT	l1 and test$3 near1 hardware	324	L2
USPT	(714/$).ccls	16278	L1

Document Number 3

Entry 3 of 3 File: USPT Jan 26, 1999

DOCUMENT-IDENTIFIER: US 5864653 A
TITLE: PCI hot spare capability for failed components

ABPL:
A system management module (SMM) for a host server system includes a system management processor (SMP) connected to a system management local bus. The system management local bus connects to the system PCI bus through a system management central (SMC). The SMC includes the main arbitration unit for the PCI bus and also includes the arbiter for the system management local bus. The SMM includes a video controller and keyboard and mouse controller connected to the system management local bus to support remote consoling of the SMM. The SMC includes logic to monitor PCI cycles and to issue error signals in the event of a system error. The SMC also isolates failed components by masking request, grant and interrupt lines for the failed device. Further, if a spare component is provided, the SMC permits dynamic switching to the spare. In addition to detecting errors and performing survival and maintenance operations, the SMC enhances system performance during normal operations by supporting master-target priority determinations to more efficiently arbitrate mastership of system busses such as the PCI bus.

BSPR:
An example of a common server host system is illustrated in FIG. 1. As shown generally in FIG. 1, one or more host processors are coupled to a system bus. The host memory couples to the system bus via a memory controller unit. The system bus, in turn, also couples to a PCI bridge which interfaces the system bus to a PCI bus. Various PCI peripherals may reside on the PCI bus. A PCI to EISA bus bridge typically couples the PCI bus to an EISA bus. Various EISA peripherals may couple to the EISA bus. In addition, and as shown in FIG. 1, an APIC bus may couple directly between the host system processor and the PCI to EISA bus bridge. Other peripherals may also couple to the system bus. Network interface cards (NIC's), for connecting the server to the PC's, may couple to either the PCI bus or to the EISA bus.

BSPR:
The video screens, resulting from a reset or failure of the server, comprise a sequence of video screen changes displayed on the host server console by the operating system, system basic input output system ("BIOS"), server application program or other system software. In particular, capture of two screen change sequences are of particular interest to a server administrator. In order to fix an existing failure or a future failure, it would be beneficial that the administrator be given the sequence of screen changes prior to server failure as well as the sequence of screen changes following a reset. Examples of server failure screens displayed on the server console are Microsoft Corp., Windows NT.RTM. "blue screen" and Novell Corp., NETWARE.RTM. ABEND message which appear on the server console when the respective operating system crashes. These screens provide information such as processor fault indicia, system software routine addresses, and pertinent system memory contents. Upon reset of the server, the power on self test ("POST") code, associated with the aforementioned operating systems,

typically performs some system diagnostic functions and displays information regarding failures detected to the server console screen. It would be desirable to capture such sequences and replaying them at a remote management site. It would also be desirable to have a remote console system which would permit diagnostic inquiries to be developed remotely which then could be transmitted to the host server system for execution. Thus, for example, tests can be run to detect open and short circuits in the system, which are the leading causes of failures. The results of these tests would then be relayed at the remote management site.

BSPR:
The problems outlined above are in large part solved by a remote communication system of the present invention. The remote communication system includes a system management module (SMM) which can be included in the server system (such as, for example, by connection to an expansion bus on the server system board). Preferably, the SMM is connected to the PCI bus, although the SMM may be located elsewhere if desired. The SMM includes a system management processor (SMP) and a system management central (SMC) control logic. The SMP and SMC connect to a system management (SM) local bus, which also connects to a VGA controller and a keyboard interface controller. The inclusion of a VGA controller on the SM local bus permits an administrator or other user to remotely monitor video data reflecting the operation of the SMM. Similarly, the inclusion of the keyboard interface on the SM local bus permits the system administrator or other user to directly access the components on the SMM (such as the SMP and SMC) by generating commands remotely for these components. The implementation of the VGA controller and the keyboard interface as part of the SMM thus permits an administrator to gather information stored by the SMM and to command the SMM to perform certain operations. In addition, the inclusion of these components in the SMM permits a user to access the server in a virtual terminal mode from a remote location.

BSPR:
The SMC of the present invention preferably includes a bus arbitration and monitor unit that preferably operates as the main arbitration unit for the PCI bus. Thus, all requests for mastership of the PCI bus pass through the SMM. The bus arbitration and monitor logic receives bus mastership requests from the various bus masters in the system and grants mastership based upon certain criteria. In addition, the arbitration and monitor logic monitors the activities of the bus master to determine if the bus master responds adequately to mastership grants. In the absence of a proper response, the arbitration and monitor logic will again attempt to grant bus mastership to the designated bus master. If the master does not perform successfully within a certain number of retries, the bus arbitration and monitor logic isolates that master from the system by masking the request/grant line to the arbitration unit and masking the interrupt lines to the system processor for that master. Thus, the defunct master cannot be granted mastership of the bus, and cannot transmit interrupts to the host system. The master therefore is effectively removed from the system until the master is repaired, or until the system is reset.

BSPR:
The SMC of the present invention also includes an IEEE 1149.1 compatible bus controller. This controller is used to run Boundary Scan tests on the host system. The SMP controls the SMC bus controller.

BSPR:
The SMM of the present invention also includes the capability to reconfigure memory and input/output space for dynamically switching to spare components without shutting down the system. If a spare is provided for a system component, the SMM dynamically reconfigures memory and I/O space in the event that the primary system component fails so that the spare assumes the address of the failed system component. In addition, the SMM functions to copy data from the failed component to the spare. During the configuration and switching, the SMM locks the appropriate bus (such as the PCI bus) to prevent any traffic on the bus

until the spare has been switched.

BSPR:
The SMM of the present invention preferably includes system management remote (SMR) units which function to monitor conditions in the host system. The SMR units couple to the SMC via an IEEE 1149.1 standard test access bus, and one or more special serial interrupt busses. The SMC and SMR units transmit signals over a serial interrupt bus (either the Granite interrupt bus (GIBUS) and/or the inter-chip interrupt bus (ICIB)) using time division multiplexing. In the event that the SMC cannot identify the source of an interrupt, the IEEE 1149.1 standard test access bus may be used to identify the source by communicating with the SMR's in the system. The SMC includes interrupt routing logic to mask certain interrupts in the event of a system failure.

DEPR:
Referring now to FIG. 2, a block diagram of an exemplary computer system 5 is shown. The computer system 5 encompasses any system which stores and retrieves files, including a file server, an application server, or any other server used in a distributed computing system. The computer system 5 preferably includes a host subsystem 25 which comprises a host central processing unit (CPU) 10 coupled to other components in the host subsystem 25 via a CPU local bus (or host bus) 35. The CPU 10 includes any data processing unit which implements a predetermined instruction set and has basic computing features such as execution units, control/timing units and registers. Exemplary CPU's include the Pentium.RTM. and Pentium Pro.RTM. processors manufactured by Intel Corp. As shown in FIG. 2, multiple CPU's 11, 12, 13 may be provided to enhance the processing capabilities of the system 5.

DEPR:
Various peripheral devices couple to the host CPU and the host bus by connection to one of a number of system peripheral busses. In accordance with one embodiment of the present invention, the bus architecture of the computer system preferably includes a 32-bit PCI (peripheral component interconnect) bus 50 and 32-bit EISA (extended industrial standard architecture) bus 60. Alternatively, the bus 60 can comprise an ISA (industry standard architecture) bus, or any other bus architecture capable of interconnecting components in a computer system. In addition, while the bus 60 is shown coupled to the PCI bus, one skilled in the art will recognize that other system configurations may be used to interface the various busses that are implemented in the system.

DEPR:
The host subsystem 25 preferably couples to the PCI bus 50. The host processor subsystem 25 includes a processor-PCI bridge 18 that connects between the PCI bus 50 and the host bus 35 to convert processor cycles to PCI cycles and vice versa to facilitate data transfers between the host bus 35 and the PCI bus 50. In the preferred embodiment, the processor-PCI bridge 18 also includes a memory controller for controlling operations to host memory 15. The memory controller 18 orchestrates the transfer of data between the host bus 35 and host memory 15. The memory controller 18 responds to various addressing techniques, and preferably is configured to support various storage cell architectures within system memory 15, including various dynamic random access memory (DRAM) implementations and static random access memory (SRAM) configurations. The processor-PCI bridge 18 is preferably implemented with an application specific integrated circuit (ASIC), but alternatively can be implemented with the 82434LX PCI/Cache/Memory Controller (PCMC) from Intel Corp.RTM.. Each of the CPU's in the host subsystem 25 preferably connect to an external cache 17. Control of the external caches 17 is provided by the processor-PCI bridge 18 or by logic in the cache 17.

DEPR:
Various peripheral devices may connect to the PCI bus 50, as will be understood by one skilled in the art. For example, server systems preferably include a SCSI hard disk controller 45 connected to the PCI bus 50 to control operations of various SCSI devices through a connector

(not shown). The various devices include hard disk drives and CD-ROM drives. Other expansion slots 47 may be provided for coupling other peripheral devices to the PCI bus 50.

DEPR:
In the preferred embodiment, a system management module (SMM) 100 preferably also connects to the PCI bus 50. The SMM 100 may be provided integrally with the computer system 5 on a system board, or may be provided as a plug-in board for connecting to the PCI bus 50 in accordance with known techniques. The SMM 100 detects failures in the computer system 5 and preferably includes the capability to correct some of those failures. As to the failures that cannot be corrected, the SMM 100 preferably is capable of isolating the source of the failure to enable the computer system to continue operation, with limited capabilities. In response to an error or failure, the SMM 100 preferably transmits an error signal to a systems administrator to initiate error analysis followed in some instances by a system reset. In accordance with the preferred embodiment, the error analysis and reset are performed remotely.

DEPR:
The system management module 100 may monitor various electrical and mechanical parameters in the system to detect system errors. Thus, for example, the SMM 100 monitors for noise in the power supply circuit, as well as for noise in other components in the computer system, determines if open and shorted connections exist in the computer system, checks for failure in the various memory devices, and detects failures in the CPU, bus bridges, hard disk controller, hard disk drives, and network interface cards (NIC's). The system management module 100 also monitors the environmental conditions of the computer system 5, such as temperature and fan operation.

DEPR:
Referring still to FIG. 2, the SMM 100 preferably couples to one or more PCMCIA connectors 95, 96 to permit the SMM 100 to communicate to other components on the network (via a NIC card) or to remote locations (via a modem). The SMM 100 also preferably includes its own power source 97 in the event of a power failure to the system. Thus, power source 97 preferably comprises a battery back-up, or other uninterruptible power supply. The SMM 100 couples to various System Management Remote monitoring units (SMR's) 71, 72, 73 in the system preferably via the IEEE 1149.1 standard test bus, and a bi-directional Granite interrupt serial bus (GI Bus) 70. In addition, the SMM 100 couples via a high speed bi-directional serial inter-chip interrupt (ICIB) bus 65 to an EISA system management remote (EISA SMR) unit 76. The SMM 100 also may connect to other buses in the system, including the APIC bus 80 and IEEE 1149.1 standard test bus 81. As one skilled in the art will appreciate, connections to some of these buses may be omitted. Conversely, in the event other buses are available in the system, the SMM 100 may connect to these buses and may transmit signals over these buses to facilitate system diagnosis and error detection.

DEPR:
Preferably connected between the PCI bus 50 and the ELSA bus 60 is a PCI-EISA bus bridge 85. The PCI-EISA bus bridge 85 preferably comprises an Intel 82375SB PCI-EISA Bridge that controls the transfer of data and address signals between the PCI bus 50 and the EISA bus 60. In accordance with normal convention, the computer system 5 includes slots 62 on the EISA bus 60 for receiving expansion boards. In one embodiment, a network interface card (NIC) 61 connects to the EISA bus 60 for controlling communications with an external local area network (not shown specifically). A data buffer 66 also connects to the data portion of the EISA bus 60 to provide an additional data bus (the X bus 90) for various components. A system flash ROM 79 receives its control and address signals from the EISA bus 60 and connects to the data portion of the X bus 90 for data transfers. Preferably, the system flash ROM 79 contains the BIOS information for the computer system 5. A floppy controller 86 also connects to the data portion of the X bus 90 for performing data transfers between a floppy disk drive (not shown) and

other components in the computer system 5. In the preferred embodiment, a system management remote (SMR) unit 76 couples to the EISA bus 60 for monitoring conditions on the EISA bus 60 and transmitting signals to the SMM 100. As will be described in more detail, the SMR unit 76 monitors the operation of the EISA bus 60 and transmits a signal to the SMM in the event of an error condition on the EISA bus 60. SMR 76 also preferably relays interrupt signals appearing on the EISA bus 60 to the SMM 100. SMR units 71, 72 and 73 connect to various system logic and busses to provide monitoring functions and data relay to the SMC. Preferably, this logic includes the host-to-PCI bridge and the PCI-to-EISA bridge, and the host bus. Thus, as shown in FIG. 2, SMR 71 couples to the host bus 35 to monitor operations on that bus. SMR 72 connects electrically to the PCI-EISA bus bridge 85 for monitoring operation of the bus bridge. Similarly, SMR 73 connects to the host bridge 18 for monitoring the operation of the host bridge. In the event of an error on the bus or bridge being monitored, the SMR unit transmits a signal to the system management module 100 via the bi-directional (GI or ICIB) bus.

DEPR:
Referring now to FIG. 3, the system management module 100 preferably includes a system management processor (SMP) 150, which preferably is implemented with the Intel 386EX processor. One skilled in the art will understand that other processors may also be used as the SMP 150. The SMP 150 preferably provides additional intelligence to the host system 5 focused primarily upon error detection and system administration to reduce the load on the host CPU 10. In addition, according to the preferred embodiment, the SM4P 150 also preferably functions as a "survival" processor in the event the host CPU 10 "hangs" or is rendered inoperable by system error. The SMP 150 thus permits access to the host computer system 5 for purposes of accessing system components to determine the source of the error. The SMP 150 also preferably detects other system errors and isolates the source of the errors to permit the host computer system 5 to continue operations.

DEPR:
The system management processor 150 preferably has the same privileges as the host CPU 10. Thus, the SMP 150 can reboot the system 5 without accessing the host CPU 10. In the event that the SMP is removed from the PCI bus 50 or the PCI bus fails, the SMP 150 can still operate and communicate with the VGA controller 175, the keyboard controller 185, the SMM NIC 192 and the SMM modem 194. In the preferred embodiment, the system management module 100 preferably isolates and utilizes separate input, output and memory maps and interrupts. The SMC 200 functions as a switch box for interrupt routing and interfacing the various busses. The SMC 200 routes keyboard and mouse interrupts individually to the system management processor 150 and to the host CPU 10.

DEPR:
Also connected to the SM bus 75 is a pair of PCMCIA slots 195, 196 for connection to PCMCIA cards, such as a network interface card (NIC) 192 or a modem 194. The inclusion of the NIC 192 on the SM local bus 75 permits the SMP 150 to perform accesses to the distributed network system even if the computer system 5 becomes inoperable. As a result, if a failure occurs in the host CPU 10, the NIC controller 61, PCI bus 50, and/or EISA bus 60, the SMM 100 still may perform accesses to the network system through a SM NIC 192 connected to slot 196 on the SM local bus 75. Similarly, the SMP 150 may send messages to a remote location via the SM modem 194, which connects to slot 195 on the SM local bus 75. Thus, the SMP 150 is capable of sending remote signals in the event that the computer system 5 crashes.

DEPR:
A local system management electrically erasable programmable read only memory (SM ROM) 160 preferably connects to the SMP 150 via control lines SC and connects to the SM local bus 75 to transmit address (SA) and data (SD) signals when prompted by the SMP 150. The local ROM 160 preferably stores power-on self-test (POST) code for the system management module 100 as well as for the rest of the computer system. The SM ROM 160 also

preferably stores test vectors for running an IEEE 1149.1 boundary scan test on the host system. The SMC contains an IEEE 1149.1 compatible bus controller for executing this test.

DEPR:
The SM ROM 160 preferably has a storage capacity of 4 Mbytes. The basic input/output system (BIOS) for the SMM 100 also preferably is stored in the system management ROM 160. The SM BIOS assumes a 4 megabyte system management DRAM 110. During initialization of the SMM 100, the SM ROM 160 writes certain patterns to the SM DRAM 110. and performs reads and compares data to determine the actual size of the system management DRAM 110. The memory mapped devices on the system management module 110 include a video controller 175, system management DRAM 110, flash ROM 160, and PCMCIA devices 192, 194. Video space size and addresses preferably are fixed, but system management DRAM 110, system management ROM 160, and PCMCIA sizes and addresses are programmable in SMC configuration registers. In addition, programmable page registers preferably are included in the SMC 200 to allow the SMP 150 to access the host memory 15 and the host I/O space. In an exemplary embodiment, each PCMCIA slot 195, 196 can be mapped into one of five memory address ranges above 64k and below 16 megabytes, starting at any 4k boundary. The SMC 200 preferably broadcasts SMP 150 memory cycles which have an address outside of the programmable address regions to the PCI bus 50. The system management power on self test (POST) program enables the default broadcast feature after determining the SM DRAM 110 size by setting an appropriate PCI definition register. Table I represents exemplary default addresses for various devices in the system management memory space:

DEPR:
For the VGA remote console support, the SMP 150 preferably reads the state of the video controller 175 at regular intervals for the purpose of duplicating the video controller state. One problem which may occur is in reading the state of the VGA controller 175 when the host CPU 10 is making an access to the video controller 175. Simultaneous accesses to the video controller 175 by the SMP 150 and the host CPU 10 presents a problem for the input/output registers used for remote switch cursor control and the like. To ensure safe accesses to the VGA registers which form part of the video controller 175, a lock signal is asserted by the SMP 150 to prevent the host from accessing the video controller 175 during accesses by the SMP. The signal is referred to as the SMC.sub.-- SLAVE.sub.-- LOCK. The SMC 200 executes lock cycles on the PCI bus by running the PCI LOCK signal until the SMP 150 executes a no lock cycle.

DEPR:
Referring still to FIG. 3, the video controller 175 preferably is representative of commonly used video controllers, such as video graphics array (VGA) controllers. The video controller 175 has associated with it a video memory for storing pixel values to be displayed on display 170. The inclusion of the video controller 175 and keyboard and mouse controller 185 on the SM local bus 75 enables the SMM 100 to function in a remote terminal mode in which a remote user can access and/or control operation of the SMM 100. The remote user may reside on the network and connect to the SMM 100 through the SM NIC 192, or may couple to the SMM through the SM modem 194 from a location removed from the network. Thus, the remote user may generate, for example, keyboard and/or mouse signals that are routed to the SM local bus 75 via SM NIC 192 or SM modem 194. The keyboard and mouse controller 185 receives the signals and encodes those signals for processing by the SMP 150. Similarly, video signals generated by the SMP (and if desired the host CPU 10) are provided to the VGA controller 175 via the SM local bus 75. The VGA controller 175 processes the signals into a pixel format for displaying on a display unit, such as display 170. In addition, the pixel signals may be transferred to a remote location via the SM NIC 192 and/or SM modem 194 for viewing at a remote location using techniques known in the art. In particular, the SMP 150 may cause pixel data to be written from video memory to the remote terminal via either the SM NIC 192 or the SM modem 194 while the host system is locked from accessing the system management local bus 75. As a possible alternative, and

assuming that the remote terminal may assert mastership of the SM local bus 75, the remote terminal may perform read cycles to the video controller 175 to read the pixel data stored in the video memory.

DEPR:
Thus, in accordance with the preferred embodiment, remote consoling support can be performed through the use of the VGA controller 175 and keyboard controller 185, even in the event of a crash by the host CPU, a PCI bus hang, and/or a loss of power. The VGA controller 175 and keyboard controller 185 preferably provide remote console support of both the SMM 100 and the host system generally, based upon signals appearing on either the PCI or the system management local bus.

DEPR:
The system management local bus 75 preferably comprises an ISA style bus which is modified to permit the SMM 100 to match VGA timing. Because the VGA controller 175 is located on the SM local bus 75, the SMP 150 also can perform write cycles to the video controller 175 for screen save programs to cut down on PCI traffic and minimize use of the host CPU 10, thereby enhancing system performance. In the preferred embodiment, the SMM 100 has bus mastership capabilities on the PCI bus 50, thus permitting the SMM 100 to run diagnostic operations on the PCI bus 50 and thus, to the rest of the system. In addition, because of the inclusion of the video controller 150 and keyboard and mouse controller 185 as part of the SMM 100, commands for the SMM diagnostic operations may originate from a remote terminal.

DEPR:
Referring now to FIGS. 3 and 4, the system management central (SMC) 200 couples to the SM local bus 75 and to the PCI bus 50. In addition, the SMC 200 also connects to various other busses that may be present in the computer system 5, including an IEEE 1149.1 standard test bus (JTAG) 81, a grant and interrupt bus (GI Bus) 70, an ICIB bus 65, and an APIC bus 80. As shown in FIG. 2, the SMC 200 couples via these various busses to system management remote (SMR) devices 71, 72, 73, 76 to monitor system operations and to detect system faults. The SMC 200 preferably is constructed as an application specific integrated circuit (ASIC). The SMC 200 provides a mechanism for converting SMP I/O to PCI configuration cycles. A second SMC may be provided on a secondary PCI bus to provide I/O APIC, PCI bus termination, and PCI arbiter capabilities to support the PCI to PCI bridge and additional PCI masters.

DEPR:
As best seen in FIG. 4, the SMC 200 preferably includes a PCI interface 205, a PCI arbiter and logic monitor 225, interrupt routing logic 235, address translation logic 240, system management registers 260, a local bus controller 270, a system management processor controller 275, and SM memory controller 280. In addition to monitoring the status of the system management remote (SMR) devices, the SMC 200 operates as a bridge between the PCI bus 50 and the system management local bus 75. The SMC (via the SM local bus controller 270) preferably tailors the cycle speed based upon the addressed peripheral on the system management local bus 75.

DEPR:
Referring still to FIG. 4, the PCI interface 205 preferably includes PCI master logic 285 and PCI slave logic 290. Transaction cycles to the system management module 100 that are made on the PCI bus 50 are processed through PCI slave logic 29(). Transaction cycles originating from the system management model 100 that are intended for targets on the PCI bus 50 (including targets on the EISA bus) are made through the PCI master logic 285. PCI master 285 and PCI slave 290 couple to the system management arbiter 210 to initiate requests to obtain mastership of the system management local bus 75. Mastership grants are routed from the system management arbiter 210 to the PCI master 285 and PCI slave 290. The PCI interface 205 also preferably couples to address translation logic 240 and local bus path logic 265.

DEPR:

The PCI slave logic 205 preferably has associated with it a force retry bit which operates under system management processor control. Assertion of the force retry bit by the SMP 150 insures that the system management module 100 will not be interrupted by PCI bus cycles during SMP accesses to local peripherals such as the VGA controller or the keyboard controller. The PCI interface 205 preferably has complete access to the entire host system when acting as a PCI master. In addition, the PCI interface controller 205 provides the system management processor 150 exclusive access to any system peripheral within the host system.

DEPR:
In the preferred embodiment, the PCI interface controller 205 preferably has the capability to run configuration cycles to configure devices on the PCI bus 50. As one skilled in the art will understand, devices on the PCI bus 50 are configured during configuration cycles by assigning an address range and an interrupt to the device. In accordance with the preferred embodiment, the SMC 200 monitors activity on the PCI bus 50 and the SMR 76 monitors activity on the EISA bus 60, and determines when a device on these busses fail. The present invention envisions the possibility of providing a back-up component on the PCI or EISA bus which is then used as a spare for the failed component. Thus, in the event that a system component fails, and a spare device is available in the system, the PCI interface controller 205 may initiate a configuration cycle to dynamically substitute the failed component with the spare. The PCI interface controller 205 preferably includes configurable memory mapping capabilities to enable PCI memory and I/O addresses to be modified. The system management registers 260 preferably are double mapped to PCI configuration space and to programmable EISA specific slots for software access. In the event that a component fails on the PCI bus or in an EISA specific slot, an alternative component that is memory mapped may be dynamically switched and activated by the system management central 200.

DEPR:
One exemplary embodiment for implementing the dynamic switching of spare components is shown in FIGS. 6 and 7. As shown in FIG. 7, the SMM 100 connects to the PCI bus 50 through address, data and control lines in accordance with normal convention. Three control signals have been shown which are generated by the SMM 100 as part of the dynamic switching capabilities. One skilled in the art will understand that these signals may he encoded using existing control lines (or combinations of lines), or may be implemented by adding additional control lines. Thus, as shown in FIG. 7, the arbiter in the SMC does not give grant to any mater other than SMC to completely lock the PCI bus 50 during the reconfiguration cycle. While in the bus lock condition all PCI masters are disabled from accessing the PCI bus absent except for the SMC on behalf of the SMP. The SMM 100 also generates a reset (RESET [n:1]) signal that is directed to the failed component 330 via an SMR to disable that device from further operations. The SMM 100 also generates an initialize spare (INIT SPARE) signal that is directed to the spare component 340 on the bus to initialize operation of the spare. As shown in the exemplary embodiment of FIG. 7, the SMM 100 includes a memory map 360 of system resources. Although the memory map is shown residing in the SMM 100, one skilled in the art will understand that the memory map may be physically located at some other location in the system. The memory map 360 preferably includes the current address range for each of the system resources, and also includes an address for each of the spare components provided in the system, plus an indication of the devices in the system for which the spare may be substituted. The memory map preferably is configurable to permit address ranges to be changed as required during switching and other system operations. Thus, if a spare is to be substituted for another device, the address range of the spare may be changed to what previously was the address range for the failed device. The address range for the failed device also is changed, or the failed device is completely disabled to prevent bus contention. As an alternative, each of the system components may be notified of the new address of the spare component, if the system components support such capabilities.

DEPR:

Referring now to FIGS. 4 and 6, an exemplary routine for implementing dynamic switching is shown. In step 301 the SMC 200 (preferably the PCI arbiter and monitor logic 225) monitors the system for failed components. If there are no component failures, the SMC continues normal monitoring and alarm operations in accordance with the principles of the present invention. If the SMC discovers that a device has failed in step 303, the SMP next checks (step 305) its configuration registers to determine if a spare device is present in the system which can assume the duties of the failed component. If no spare is available, the SMP isolates the failed device and cuts the request, grant and interrupt lines for the failed device (step 307). If the failed device is located on the EISA bus, the SMP initiates a slot specific reset in an attempt to reset the failed component. The reset is transmitted to the failed components via the SMC, through the SMR. If the device continues to fail and causes the EISA bus to fail, the SMP may disable the PCI-EISA bridge 85 to prevent transactions between the PCI bus and the EISA bus.

DEPR:
Referring still to FIG. 4, address translation logic 240 functions to translate address signals on the PCI bus 50 and the system management local bus 75 to correspond to appropriate bus protocols. The PCI bus 50 preferably is 32 bits wide and the SM local bus 75 is either 16 bits or 8 bits wide. The address translator 240 takes SM addresses and converts these addresses to PCI addresses, and similarly, converts PCI addresses to appropriate SM address values. A local bus data path module 265 contains all the logic that is used to capture data from the SM bus 75 and for multiplexing data from the PCI bus 50 to the SM bus 75.

DEPR:
The address translation logic 240 is capable of redirecting system ROM so that the SMC can get the host's CPU to use the system management ROM 160 instead of the host system ROM 15. The address translation logic 240 preferably provides support for memory mapped I/O from the PCI bus 50 to the system management local bus 75 and from the system management local bus to the PCI bus. Thus, the address translation logic 240 permits address and space translation of PCI memory cycles to system management module input/output and/or memory space. This permits access to I/O locations on the system management module 100 that are already occupied by other devices in the host system I/O space. Similarly, the address translation logic 240 provides address and space translation of SMM memory cycles to PCI I/O and/or memory space. This permits access to I/O locations on the host system that are already occupied by other devices in the SMM I/O space. The address translation logic 240 also directs SMM cycles to either the SM bus or to the PCI bus depending upon the address.

DEPR:
Referring still to FIGS. 3 and 4, a signal ADFLT for tri-stating the address and data outputs of the SMP 150 is provided by the SM arbiter 210 to the SMP 150. As mentioned above, there are two possible masters on the SM local bus 75, the SMP 150 and a PCI bus master. A request from the SMP 150 is indicated by a processor request (PREQ) signal asserted by the processor controller 275 to the SM arbiter 210. A request from a PCI bus master is indicated by a local bus request (ISLREQ) signal generated by PCI slave logic 29() in the PCI interface controller 205 to SM arbiter 210. The various cycles executable by the modules in the SMC 200 preferably include a refresh cycle to SM DRAM 110, a PCI cycle to the video controller 175, a SMP cycle to SMC registers 260 (which include various memory and I/O map registers and configuration registers), a PCI cycle to the SMC registers 260, an SMP cycle or a PCI master cycle to the SM memory 110; an SMP cycle or a PCI master cycle to the SM ROM 160, a SMP cycle or a PCI master cycle to the SM local bus 75, a SMP cycle or a PCI master cycle to the IEEE 1149.1 standard test bus controller (JTAG) logic 220, and a SMP cycle to the video controller 175. The SMP 150 also provides a processor read/write (PWR) signal to the SM arbiter 210 indicating processor read or write cycles. When a PCI master is requesting a cycle to a target in the SMM 100, the SM arbiter 210 asserts a hold (HOLD) signal to the SMP 150, which the SMP 150 acknowledges by asserting the hold acknowledge (HLDA) signal. Cycles on

the SM local bus 75, other than to the SM memory 110, requested by the SMP 150 or a PCI bus master are generated by the local bus controller 270. The local bus controller 270 controls accesses to the video controller 175 and the SM ROM 160.

DEPR:
If the SM arbiter 210 detects a request for the SM memory 110 from the SMP 150 or from a PCI bus master, it asserts a RUNMEMCYCLE signal to the memory controller 280. Memory requests from the SMP 150 include read and write cycles as well as memory refresh cycles. As noted earlier, the SMP 150 preferably includes logic to arbitrate between refresh and other memory cycles. In response to assertion of the signal RUNMEMCYCLE, the memory controller 280 provides the necessary control signals to the SM memory 110 to perform the requested operation. When a memory operation is completed, the memory controller 280 asserts a memory cycle done (MEMCYCLEDONE) signal back to the SM arbiter 210. In response, the SM arbiter 210 negates the signal RUNMEMCYCLE. If the SM arbiter 210 detects a SMP-to-PCI cycle, it asserts a signal RUNIMSTCYCLE to PCI master logic 285 in the PCI interface controller 205. In response to the assertion of the signal RUNIMSTCYCLE, the PCI bus master 285 obtains control of the PCI bus 50 to run the desired cycle. Once the PCI cycle completes, the PCI master 285 responds by asserting the signal IMSTCYCLEDONE,. A few clock periods later, the SM arbiter 210 deasserts the signal RUNIMSTCYCLE. When the SM arbiter 210 negates the RUNIMSTCYCLE signal, a signal D.sub.-- IMSTCYCLEDONE in the PCI master logic 285 is negated one clock (PCICLK) period later, which causes the signal IMSTCYCLEDONE to be negated.

DEPR:
When the PCI or SM local bus cycles have completed, the SM arbiter 210 drives a signal PDONE to the processor controller 275 to indicate the completion of a SMP requested cycle, or asserts a signal ISLDONE to the PCI slave logic 290 to indicate the completion of a PCI bus master requested cycle.

DEPR:
Referring now to FIG. 4, the PCI arbiter and monitor logic 225 couples to the PCI bus 50 and functions as the main PCI arbitration unit for the PCI bus. Thus, to the extent that other PCI arbiters are present in the computer system, those arbiters preferably are disabled in favor of the arbiter and monitor logic 225 in the SMC 200. PCI arbiter and monitor logic 225 preferably couples to the system management registers 260, and includes a clemency counter 227 and an event counter 229. Counters 227 and 229 may be integrated as part of PCI arbiter and monitor logic 225, as will be understood by one skilled in the art.

DEPR:
The PCI arbiter and monitor logic 225 monitors the PCI bus 50 to determine if signals appearing on the PCI bus are consistent with PCI protocol. If signals appear out of sequence, the PCI arbiter and monitor 225 generates an error signal which preferably is transmitted to the system management processor 150 and host processor 10. Preferably, the PCI arbiter and monitor 225 also monitors the length of time of certain transactions, including (a) the time before FRAME# is asserted after REQ# and GRANT# are asserted on the PCI bus; (b) the time before data is returned or provided after the assertion of the FRAME# signal on the PCI bus; and (c) the length of time that the PCI bus is held after the REQ# signal is negated. Measurement of the response time preferably is performed by counter 229. Counter 229 may comprise a plurality of counters for each of the different time out values or a single counter may be used with additional processing performed by the PCI arbiter and monitor logic 225. Alternatively, the PCI arbiter and monitor 225 may include one or more internal counters for performing the time-out functions.

DEPR:
The PCI arbiter and monitor 225 preferably captures PCI bus master lock commands and addresses upon error occurrence on the PCI bus. In the event of a failure on the PCI bus which causes the PCI bus to "hang,"

the PCI arbiter and monitor 225 determines how far the current cycle has progressed. Information regarding the current bus cycle may be stored either in the PCI arbiter and monitor logic 225 or in the system management registers 260. In the event of a failure, appropriate registers bits are read back indicating the signals that had been received during the current bus transaction. Thus, for example, the PCI arbiter and monitor 225 may monitor the PCI bus for the presence of the REQ#, GRANT#, FRAME#, DEVSEL#, IRDY#, TRDY#, and various other PCI bus signals generated during the current cycle, and in response to detection of the PCI signals, may provide an appropriate signal to the system management registers 260 to load appropriate register bits. In the event of a failure, the processor controller 277 accesses the system management registers 260 to read back the appropriate register bits for the current cycle. In addition, the PCI arbiter and monitor logic 225 also preferably stores a value indicating the component that has mastership of the PCI bus, the current address and the cycle type when the PCI bus fails. For parity failures detected by the PCI arbiter and monitor 225 or other SMC components, the address or data causing the failure preferably is latched and stored in the system management registers 260.

DEPR:
The PCI arbiter and monitor 225 also monitors the operation of components residing on the PCI bus as the main PCI arbitration unit. In the preferred embodiment, the PCI arbiter and monitor logic 225 includes the capability of generating an error signal if several accesses are made to a PCI component without a response. In this vein, and referring now to FIG. 8, the PCI arbiter and monitor logic 225 tracks the activities of the various PCI bus masters. In step 282, the PCI arbiter logic 225 determines if PCI masters have requested ownership of the PCI bus. If more than one request is received, the arbitration logic 225 cheeks the priorities of the requesting masters (step 284), and in step 286, issues a grant to the master with the highest priority. The PCI arbiter and monitor logic 225 then monitors the PCI bus (step 288) to determine if the master granted ownership of the bus has started a cycle. If the master does not initiate a cycle within a specified time period (as determined by event counter 229, for example), the arbiter logic 225 checks the count in the clemency counter 227 for that master. If the count in the clemency counter exceeds a predetermined threshold value x (step 292), the arbiter logic 225 disables the master in step 294 by masking the grant and interrupt lines for that master. If the count in the clemency counter 227 is less than x, the clemency counter is incremented for that master (step 296). After other master requests are serviced in step 298, the faulty master is retried in step 299.

DEPR:
Thus, if a PCI bus master fails to respond to a mastership grant by the PCI arbiter and monitor 225, its grant is removed if there is no other request pending as part of a normal re-arbitration. On the first grant removal, a counter (such as event counter 229) is activated and the current master number is recorded in an appropriate register in system management registers 260 or in a latch in the PCI arbiter and monitor logic 225. Subsequent retries to the same master cause a counter to be incremented. If the master responds, the clemency counter 227 is reset. If the count in the counter becomes higher than a pre-established maximum value x an error signal is asserted to the system management processor 150. In addition, the PCI arbiter and monitor logic 225 disables the ability of that bus master to obtain mastership of the PCI bus. In the preferred embodiment, a register is provided with bits dedicated to the availability of various bus masters on the PCI bus. If a failure is detected for a particular bus master, a bit is set in a dedicated bit associated with that bus master device. When a mastership request is received by the PCI arbiter and monitor logic 225, the error register is examined before mastership is granted by the PCI arbiter and monitor logic 225. While one clemency counter is shown, one skilled in the art will understand that a plurality off clemency counters may be implemented if desired.

DEPR:

The PCI arbiter and monitor logic 225 also preferably is capable of selecting and storing PCI bus cycles for subsequent examination. The PCI arbiter and monitor logic 225 includes a circular buffer, preferably constructed as a FIFO (first in, first out) register. In addition, the PCI arbiter and monitor logic 225 operates in association with registers in the SMC registers 260, including a bus capture register, a holding register and a cycle progress resister. The PCI arbiter and monitor logic 225 causes cycles to be stored in the registers including input/output cycles, memory cycles, configuration cycles and special cycles. Values in the registers can be read by the system management processor in the case of a bus fault on the PCI bus. In addition, the bus capture register and the circular buffer preferably are not cleared during SMC resets. The bus holding register preferably comprises an intermediate register located between the bus capture register and the circular buffer.

DEPR:
The cycle progress registers are used to indicate the progress of the current transaction on the PCI bus. An example of the tracker registers is shown in Table IV:

DEPR:
In the preferred embodiment, the PCI arbiter and monitor logic 225 is capable of determining bus master usage, total bus master access latency, total target latency, bus master access latency, the number of retries to a particular PCI device, the number of PCI bytes transferred in a particular cycle, the amount of time that the PCI master holds the bus after the negation of the REQ signal, and various other performance criteria.

DEPR:
The system management registers 260 preferably are accessible either by I/O or PCI configuration cycles from the PCI bus 50 or by I/O cycles from the system management local bus 75. The SMC 200 preferably is capable of individually disabling any input/output or memory region from either the PCI side or the system management side, and can also disable the source of any detected error. To expand the system management processor address base from 64 megabytes to four gigabytes, the SMC 200 translates SMP addresses using segment registers. The SMC 200 is initialized once the power source (either system power or battery backup 97) is confirmed as acceptable (battery ok is asserted). After power up, the SMC 200 determines clock phases and asserts a synchronous reset to the SMP 150 and to the SMR's in the system.

DEPR:
The PCI arbiter and monitor logic 225 preferably receives bus mastership requests and assigns or grants mastership of the PCI bus 50. If a bus master fails, the arbitration and monitor logic 225 deactivates the failed device. Whether a master has failed can be determined using various methods. One method which can be used to determine whether a master has failed is through software. Thus, a software diagnostic utility can be run periodically on the system to monitor the response of each device on the bus. If the arbiter and monitor logic 225 determines that a device has failed, it can ignore all mastership requests from that device, and refuse to grant mastership to the failed device. In addition, the SMC 200 can mask the interrupt requests from the failed master through the interrupt routing implementation shown in more detail in FIG. 5. The SMC 200 receives all of the interrupts in the system through various busses and passes the interrupts to the SMP 150.

DEPR:
A number of SMR units are dispersed through the system to provide additional monitoring capabilities for the SMM 100. The SMR's communicate with the SMC 200 via high speed bi-directional serial busses (either the GI bus or the ICIB bus) and the IEEE 1149.1 standard test bus. Time division multiplexing is used on the GI bus or the ICIB bus, with the SMC driving high, and the SMR's driving low. The devices on the serial bus are synchronized to a particular counter.

DEPR:
As will be understood by one skilled in the art, the EISA controller asserts RESDRV as a response to an EISA bus error. The bus monitor logic 725 intercepts this signal and generates an interrupt to the SMP 150 (FIG. 3) via the IBUS (which for the EISA SMR 76 is the ICIB bus). The SMR signals this interrupt to the SMC 200 (FIG. 3) by asserting In interrupt on slot 2 of the ICIB bus. The SMC 200 then passes this interrupt to the SMP. After receiving the RESDRV interrupt, the SMP 150 determines the MAK# signals on the EISA bus by reading the status of a MAK# latch register in bus monitor logic 725. After determining which master had control of the bus when RESDRV was asserted, the SMP 150 causes a slot-specific RFSDRV to be sent to the offending master.

DEPR:
The JTAG slave logic 705 is based upon the Compaq JTAG slate standard cell design; however, the standard cell design is modified to permit the JTAG lines to be used for communication to all registers within the SMR. The SMR is connected to the SMC and the other SMR's via the IEEE 1149.1 compatible standard test bus (JTAG bus [4:0]) set up in a ring configuration. The slave logic 705 includes a JTAG multiplexer (not shown specifically) that expands the JTAG chain, allowing the JTAG slave to select any of six JTAG chains with which the SMR interfaces.

DEPR:
Thus, in summary, if a failure occurs on the EISA bus, the EISA monitoring SMR can reset the EISA bus, or can reset slots on the EISA bus. If problems continue on the EISA bus, the PCI-EISA bridge can be disabled by the PCI arbiter and monitor logic 225. The error is then reported via the SM modem or SM NIC to a remote site.

DEPR:
Thus, as shown in the flow chart of FIG. 9, the arbitration and monitor logic 225 determines in step 401 whether a mastership request appears on the PCI bus. If a mastership request does appear, the arbitration logic 225 awards mastership in step 403 based upon either a round robin approach, or based upon priorities assigned to specific masters, in accordance with conventional techniques. After mastership is granted, the arbitration and monitor logic 225 determines the target of the current cycle in step 405. In step 407, the arbitration and monitor logic 225 checks the current master-target combination to determine if the combination has been determined to be a high priority combination, as specified in the high priority registers (see for example TABLE V).

DEPR:
Thus, after granting mastership of the PCI bus, the arbitration and monitor logic 225 preferably monitors the target address. Once the target address is determined for the current transaction, the arbitration and monitor logic 225 compares the master and target address with the combinations listed in the priority registers (Table V). If a match is found, that master is elevated to the highest priority as long as that master continues to request mastership of the bus. One skilled in the art will understand that more or less than four registers may be provided to specify greater or fewer than four high priority master-target combinations. In addition, one register may be used to encode more than one master-target pair depending upon the register size and the number of bits required to specify the address of the master and target. Further, although the present invention is discussed by referencing the PCI bus, one skilled in the art will understand that the present invention can be used on other bus systems as well which have bus mastership capabilities.

DEPR:
In the hot master/target scheme shown in FIG. 11B, the M2 master has been determined to be addressing a target that has been predetermined to be a high priority master/target pair (such as, for example, a CPU master and a video controller slave). Thus, the arbitration unit returns to the high priority M2 Master after completing a mastership cycle with another master. Thus, as shown in FIG. 11B, after a M1 Master cycle, M2 is granted mastership. The next master cycle is granted to M3, with M2

again given a grant in the next master cycle. In this fashion, the high priority master is given every other mastership cycle which addressing a high priority target.

DEPR:
Referring still to FIG. 4, interrupt routing logic 235 receives interrupt requests from various components in the system. Thus, interrupt routing logic 285 receives interrupt requests originating on the PCI bus 50, interrupt requests originating on the EISA bus, PCMCIA interrupt requests, mouse and keyboard interrupt requests, and other requests made on or transmitted via the APIC bus. In addition, interrupt routing logic 235 preferably receives signals from the SMR's 71, 72, 73 (FIG. 2) via the GI bus 70. Interrupt logic 235 also receives a signal from the SMR 76 via the ICIB bus 65. Both the GI bus 70 and the ICIB bus 65 preferably are bi-directional serial busses which use time division multiplexing to schedule transmissions. If the source of an interrupt cannot be determined, the IEEE 1149.1 standard test bus is used to assist in locating the source of interrupts. As shown on FIG. 4, signals on the APIC bus are routed through APIC I/O 215. An IEEE 1149.1 compatible bus controller 220 Functions to couple the SMC 200 to the IEEE 1149.1 standard test bus. The IEEE 1149.1 compatible bus controller 220 couples to command logic 250. In accordance with the preferred embodiment, command logic 250 initiates cycles on the IEEE 1149.1 standard test bus 81 to assist in locating the source of interrupts. In addition, other control signals may be transferred through the IEEE 1149.1 standard test bus 81. Thus, for example, in instances where the PCI bus is inoperable, the SMC 200 still may perform cycles to other system components via the IEEE 1149.1 standard test bus. Thus, in a failed system, the SMC 200 can interrogate other system components through test vectors stored in EEPROM, which are generated using Texas Instrument Asset tools and Victory software. The IEEE 1149.1 compatible bus controller 220 preferably includes both a master block and a slave block to permit two SMC units to communicate if more than one SMC unit is provided in the system. Each block has its own set of JTAG pins so that the IEEE 1149.1 standard test bus 81 may be used to configure and control the SMR's, and to write to other devices on an IIC bus (or other bus) via a JTAG-to-IIC bridge in the SMR. The SMC IEEE 1149.1 standard test controller 220 is accessible by both the SMP 150 and the host CPU 10.

DEPR:
Referring now to FIGS. 4 and 5, the SMC 200 collects remote interrupt requests for the SMP 150 and the host CPU 10 to read. Several SMP state machines detect and report abnormal conditions by setting error bits. Once the SMC detects an error condition, it interrupts the SMP. If a catastrophic failure is detected, the SMC asserts a system error (SERR) signal which is connected to the EISA controller ESC to produce an NMI signal. The SMC stores the first error detected in the First Error Register and all detected errors in the "Error Register" to allow a more intelligent failure analysis.

DEPR:
The interrupt routing scheme according to the preferred embodiment is shown in FIG. 5. All interrupts from the SMM components (such as PCMCIA adapters 194, 195, SMP 150, keyboard and mouse controller 185, and SMR 325) are routed through SMC 200. The SMC 200 also receives interrupt requests from SMR's 71, 72, 73 on GI bus 70, and from SMR 76 on ICIB bus 65. The SMC 200 also receives EISA interrupt signals on the APIC bus 80. Interrupts are routed from the SMC 200 to SMP 150 on APIC bus 80. In one embodiment of the invention, a second SMC 300 is provided on a secondary PCI bus 350. The second SMC 300 couples to the APIC bus 80 and ICIB bus 65 to assist in processing interrupt requests from the secondary PCI bus.

DEPR:
In the preferred embodiment of FIG. 4, the system management processor 150 dynamically ascertains the JTAG topology as part of system initialization. In accordance with known techniques described in the IEEE 1149.1 standard test access port description, software running on

the SMP 150 performs an ID scan to determine what chips and XN hat JTAG topology are present in the system. After determining that the system contains one or more SMR's, which serve as JTAG bridges, the software can use the SMR's to connect to other JTAG chains and perform an ID scan to determine the topology of this chain. As shown in FIG. 4, the SMC 200 preferably includes a IEEE 1149.1 standard test controller 220. The 200 allows the SMP 150 to run a boundary scan of all scanable components in the computer system. The IEEE 1149.1 compatible bus controller 220 includes a response FIFO set which allows the SMP 150 to pipeline several commands to the JTAG port. The SMP 150 is capable of reporting the various conditions of the computer system to the operator via the computer screen display or through the modem or local area network connected through the PCMCIA slots 195, 196.

DEPR:
The IEEE 1149.1 compatible bus controller 220 interfaces the SMC 200 to the IEEE 1149.1 standard test bus 81. Each of the SMR's 71, 72, 73, 76 preferably includes a register to control multiplexing in the SMR. The register preferably is included in the JTAG data chain to permit modification of the state of the multiplexer. A JTAG master device permits the IEEE 1149.1 standard test bus to be used for testing and management communications in accordance with the preferred embodiment. Thus, the IEEE 1149.1 standard test bus is used to perform serial communications between the SMR and SMC as well as function as a test in accordance with normal techniques. In the preferred embodiment, PCI masters can scan the SMC and thus the SMP via the IEEE 1149.1 standard test bus. Thus, for example, the host CPU can scan the system management module via the IEEE 1149.1 standard test bus if desired.

DEPR:
Referring still to FIG. 4, APIC I/O 215 functions to interface the SMC 200 to the APIC bus 80. In the preferred embodiment, the SMC 200 implements a remote read operation used in the I/O APIC which permits the SMC 200 to read the host processor's internal states as part of a diagnostic feature. The I/O APIC logic 215 also preferably supports interrupt routing from the SMC to the host on the APIC bus 80 in the event that the host CPU does not support APIC protocol.

CLPR:
2. A computer system as in claim 1, wherein the computer system includes a memory map of resources on the PCI bus, and the system management module reconfigures the memory map when switching the spare component for the primary component.

CLPR:
5. A computer system as in claim 4, wherein the system management module further includes system management memory connected to said system management local bus, and wherein a memory map of system resources on the PCI bus are maintained in the system management memory.

CLPR:
12. A server as in claim 11, wherein the system management module includes a memory controlled by said system management central, said memory including a map of resources on the PCI bus.

CLPV:
a bus bridge connecting said host bus to a PCI bus;

CLPV:
a system management module connected to said PCI bus, said system management module includes a system management processor;

CLPV:
a primary component connected to said PCI bus for performing specified operations on the PCI bus;

CLPV:
a spare component connected to said PCI bus;

CLPV:
wherein the system management module includes monitor logic for detecting failure of the primary component, and in response, the system management module locks the PCI bus to dynamically switch the spare component for the primary component.

CLPV:
a PCI bus bridge coupling said host bus to a PCI bus;

CLPV:
a primary component connected to said PCI bus for performing specified operations;

CLPV:
a spare component connected to said PCI bus, said spare component having the capability to perform at least some of the operations of said primary component;

CLPV:
a system management module connected to said PCI bus, said system management module including:

CLPW:
a system management central unit interfacing a system management local bus to the PCI bus, said system management central unit coupling to the system management processor through the system management local bus;

CCOR:
714/2

CCXR:
714/7